Analyzing Car Prices in Serbia

Nilson Gao (1008074049) - Background and Diagnostics
Rebecca Han (1007808124) - Data Analysis Ben Wang (1007727024) - Models
Vedat Goktepe (1007798661) - Validation and Conclusion

12-03-2024

Research Context

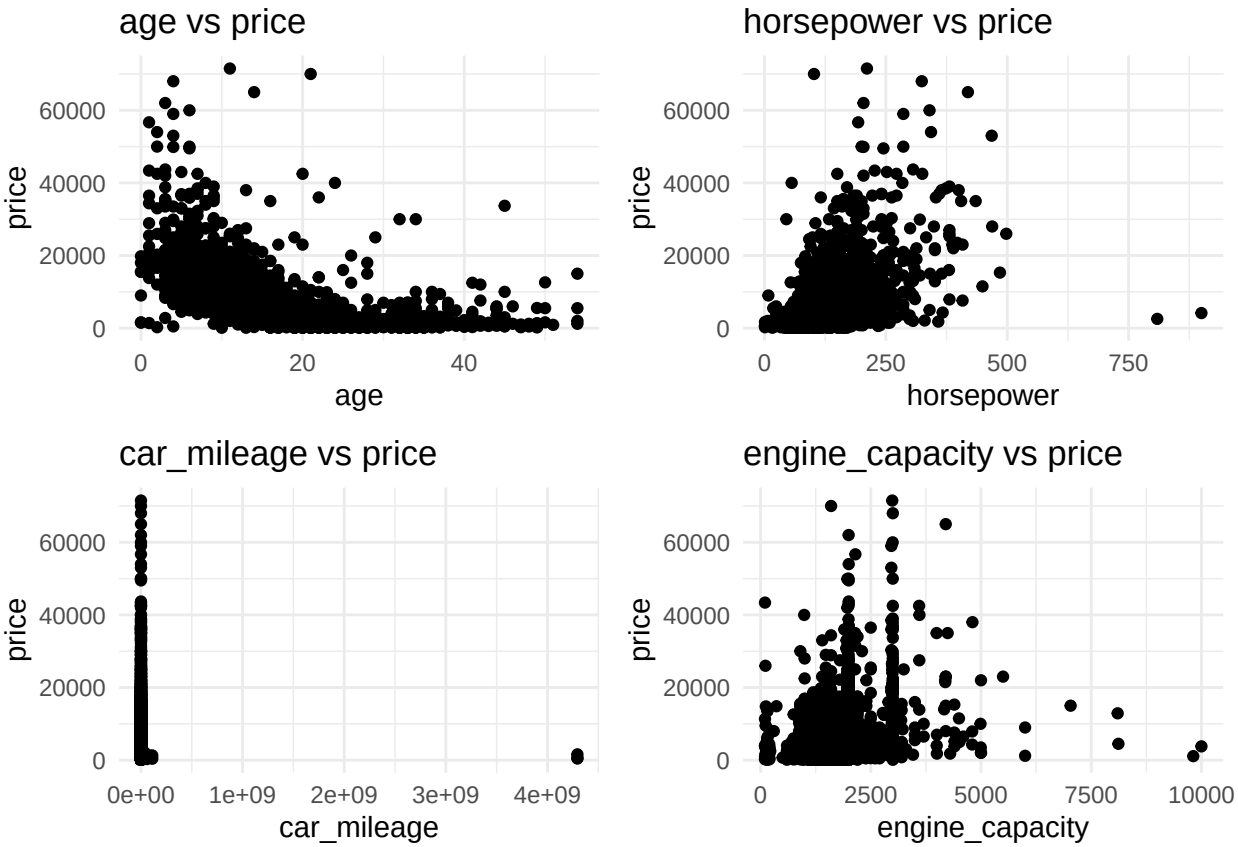
The objective of this research is to investigate the intrinsic factors that affect Price of Cars in Serbia based off detailed car listings provided by an online marketplace using a dataset from 2024 to examine variables like age, mileage, horsepower, and engine capacity. This research aims to help consumers, dealerships, and manufacturers better understand the car market dynamics. For Serbia, a country with a growing used car market, understanding these factors provides valuable insights. For example: Age: Do older cars lose value faster? Mileage: Does more usage lower prices significantly? Horsepower and engine capacity: Are they key drivers of value? This knowledge can guide buyers in making informed choices and sellers in pricing their vehicles competitively.

Exploratory Data Analysis

```
# read data file, published in 2024 on
# https://www.kaggle.com/datasets/mmakarovlab/serbia-car-sales-prices?resource=download
car_price_data <- read.csv("serbia_car_sales_price_2024.csv")
validation_data <- read.csv("serbia_car_sales_price_2024.csv")
```

Before we begin investigating, we notice that there are some issues with the data. Some rows are missing values under certain variables (i.e. #2, #233, #1705, etc.), and some variables are hard to work with. Knowing a car's **year** might be less informative than knowing its age, so we made a new column containing values for 2024 − Year called **age**. A car's **horsepower** is significant, but it's hard to use that data when it's given as two values in the format HP (kW), so we keep only the HP metric. Additionally, some variable names are hard to work with because of length or how it might interfere with R code, such as **car_mileage, km**, so we made those easier to process as well. As for the missing values, when we analyze the significance of a variable, we'll make sure to exclude rows where values for that variable are empty.

Now, we want to check on which variables are good predictors. For the continuous variables, we first plot scatter graphs for each variable against car price:

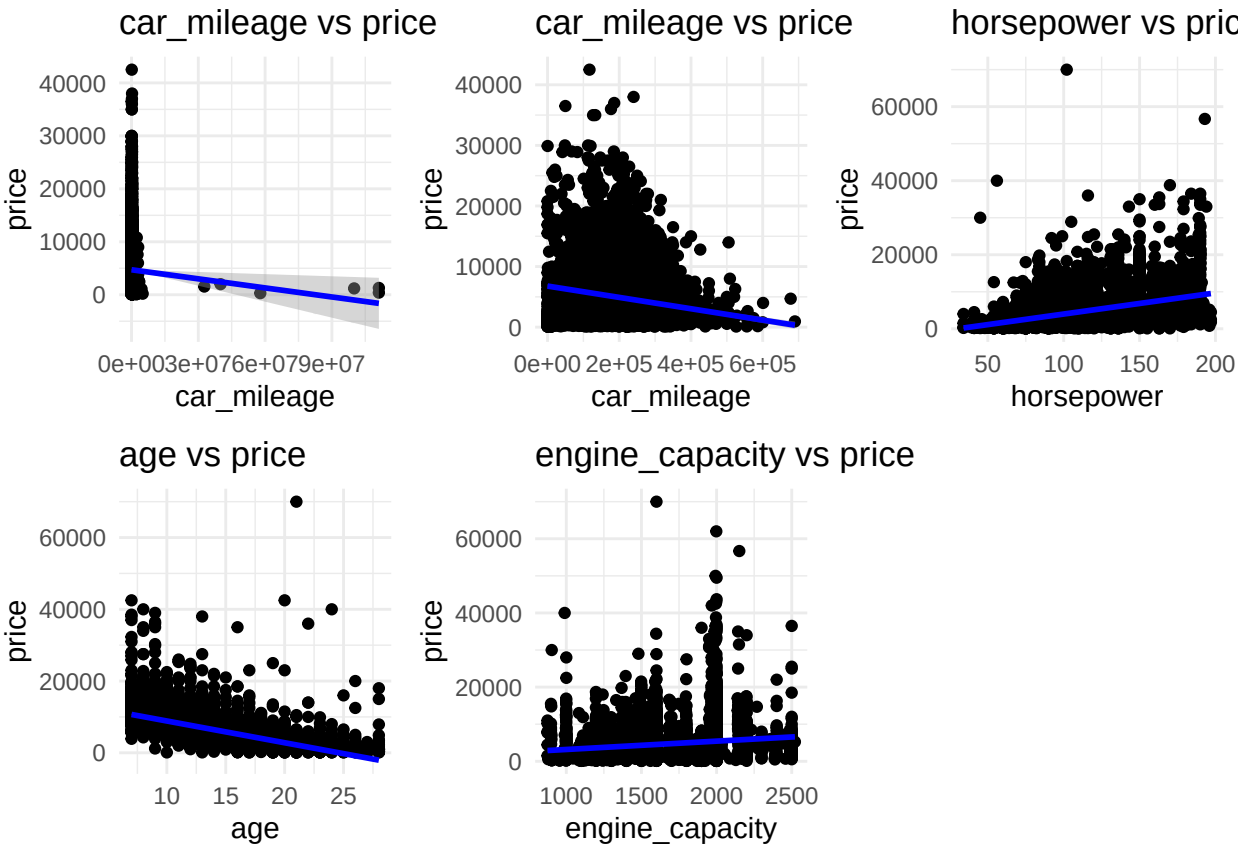


We notice that there are some influential points (and possibly leverage points) on the car_mileage predictor. Let's

get rid of those using hat values and semistudentized residuals to detect which ones are too high:

After outlier cleaning, let's check on the scatter plots again to see if these continuous variables have a significant influence on the car price:

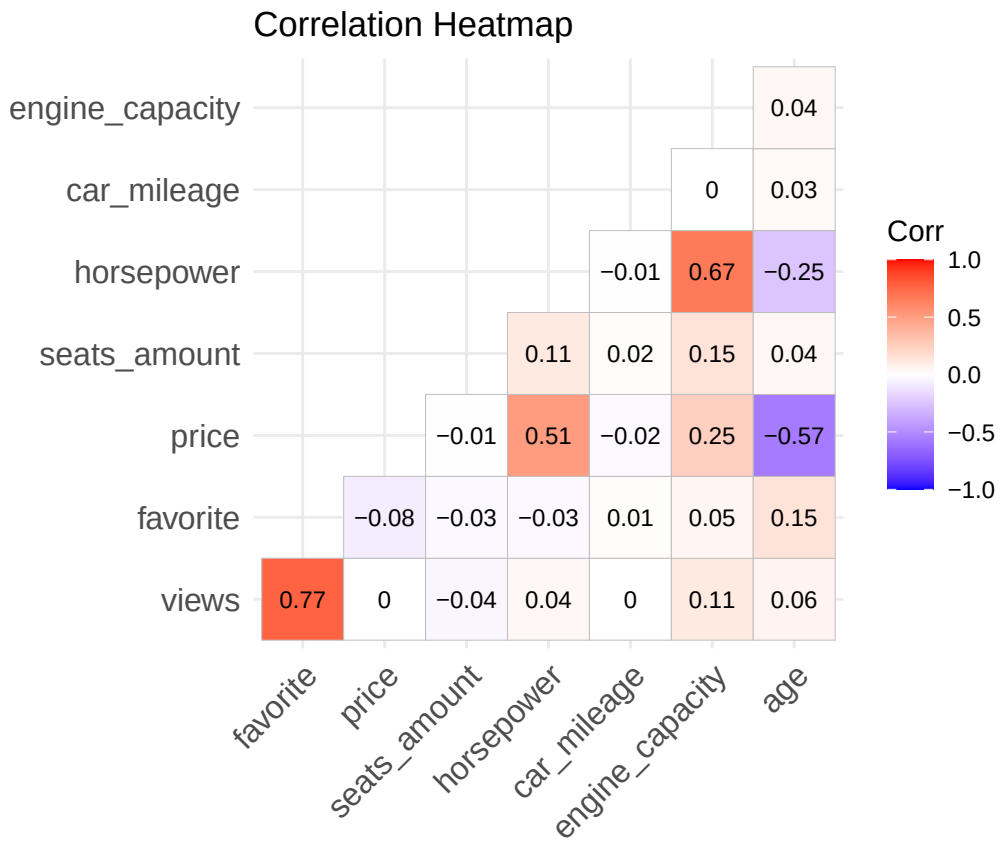
```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



Now we can clearly see that price tends to decrease as age or mileage increases, and price tends to increase as horsepower or engine capacity increases. This gave us a lot of confidence in our dataset, since this is about what we logically expected to happen based on prior knowledge of how car pricing works.

Now moving on to the categorical variables.

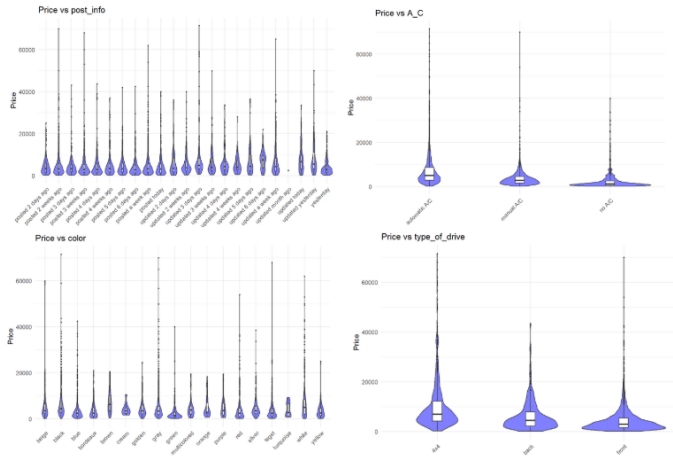
We need to make sure that there's no correlation between the different categorical variables.

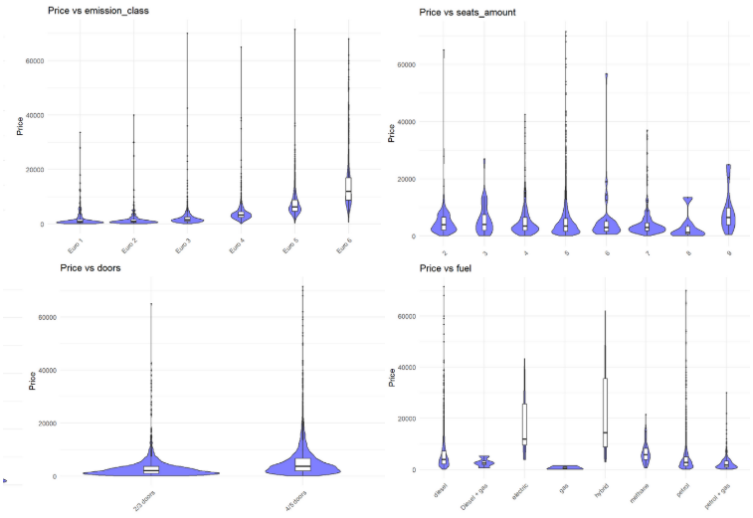


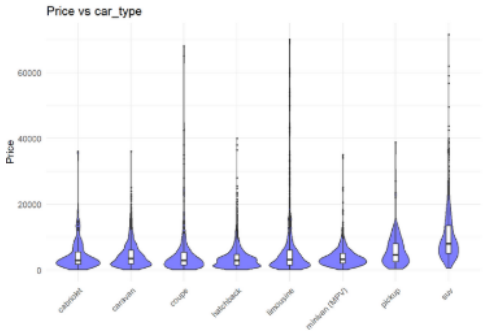
Before we delve further into data analysis, we notice that there's some information in our dataset that is unlikely to be relevant, such as how many views or favourites the car posting gets, or the date of which it was posted. However, we need to run a t-test to make sure that those variables indeed do not have any influence on the final price of the car. We will be removing views and favourites from our model for two main reasons: we wish to know the main intrinsic factors that help determine price, and because views and favourites reflect market interactions, which can be influenced by listing quality, pricing strategy, and marketing, rather than the car's characteristics.

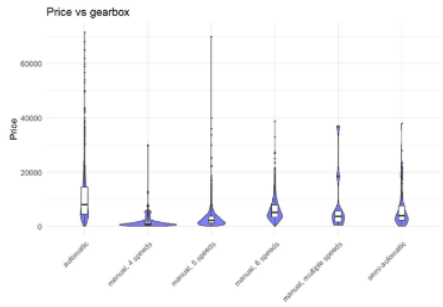
Categorical Variable Analysis

screenshots of violin charts for formatting purposes:





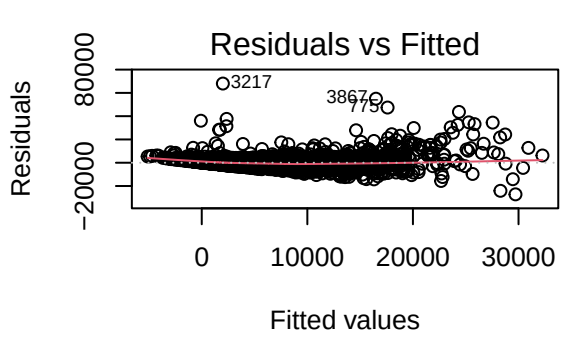


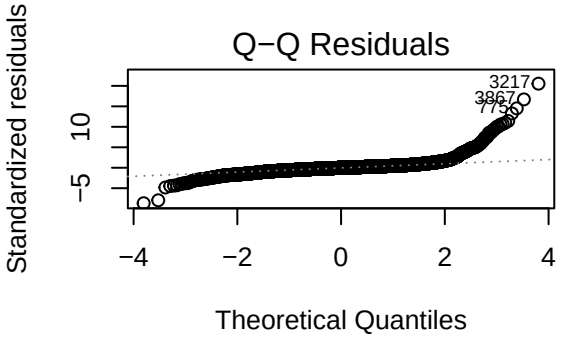


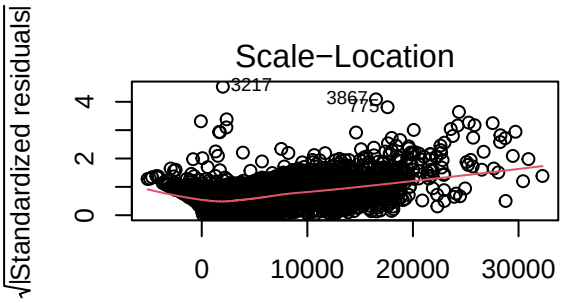
Main Effect Model

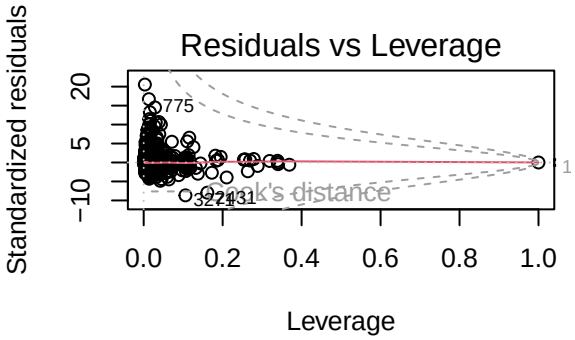
```
##                      GVIF Df GVIF^(1/(2*Df))
## views            2.679640  1        1.636961
## favorite         2.753326  1        1.659315
## post_info        1.734536 22        1.012595
## year             4.454135  1        2.110482
## A_C              2.218061  2        1.220375
## emission_class   5.907211  5        1.194368
## seats_amount     3.219628  7        1.087106
## horsepower       2.997857  1        1.731432
## color            1.720354 17        1.016085
## car_mileage      1.008023  1        1.004003
## engine_capacity 2.526592  1        1.589526
## type_of_drive    2.532844  2        1.261543
```

```
## doors           1.797410  1        1.340675
## fuel            1.994162  7        1.050537
## car_type        8.257759  7        1.162760
## gearbox         3.532257  5        1.134502
```









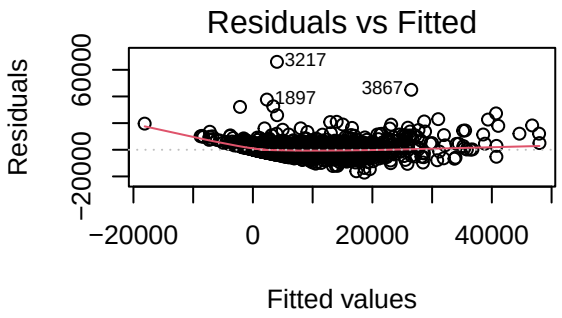

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## price ~ views + favorite + post_info + year + A_C + emission_class +
##     seats_amount + horsepower + color + car_mileage + engine_capacity +
##     type_of_drive + doors + fuel + car_type + gearbox
##
## Final Model:
## price ~ views + favorite + post_info + year + A_C + emission_class +
##     seats_amount + horsepower + engine_capacity + type_of_drive +
##     fuel + car_type + gearbox
##
##
##             Step Df    Deviance Resid. Df  Resid. Dev      AIC
## 1                                    6994 76883584777 114802.9
## 2       - color 17 207259338.9      7011 77090844115 114788.0
## 3 - car_mileage  1    370832.3      7012 77091214948 114786.0
## 4       - doors  1   1250094.6      7013 77092465042 114784.1
```

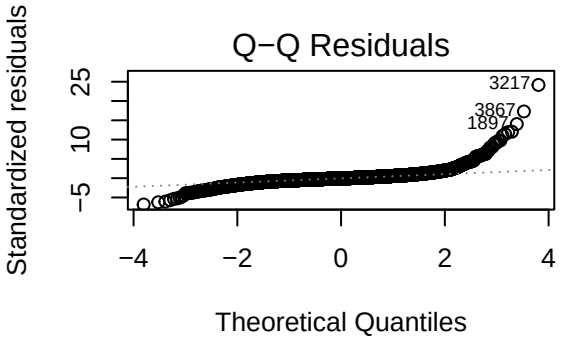
We are worried this model may not capture any interaction between the variables. So, let's run an interaction model.

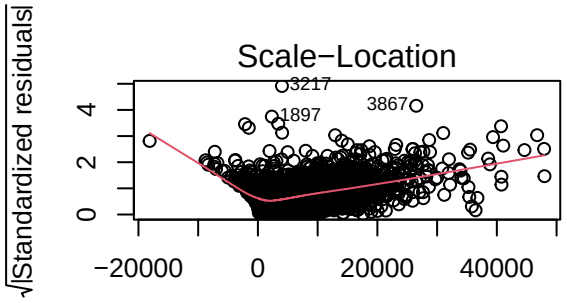
Model with Interaction Terms

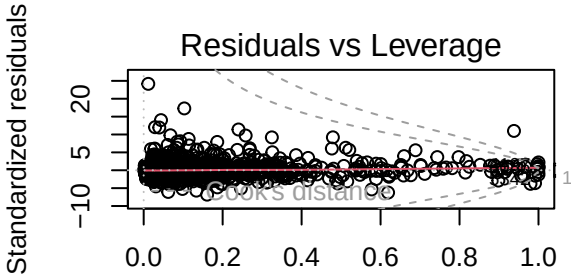
```
##                    GVIF Df GVIF^(1/(2*Df))
## views          2.668527  1        1.633563
## favorite       2.746790  1        1.657344
## post_info      1.622043 22        1.011054
```

```
## year             4.376946  1        2.092115
## A_C              2.133579  2        1.208585
## emission_class   5.575864  5        1.187493
## seats_amount     2.883749  7        1.078584
## horsepower       2.979929  1        1.726247
## engine_capacity 2.513466  1        1.585392
## type_of_drive    2.508960  2        1.258559
## fuel             1.924037  7        1.047854
## car_type         6.236129  7        1.139671
## gearbox          3.433421  5        1.131287
```










```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## price ~ age + car_mileage + engine_capacity + horsepower + post_info +
##     A_C + seats_amount + color + type_of_drive + doors + fuel +
##     car_type + gearbox + age:post_info + age:A_C + age:seats_amount +
##     age:color + age:type_of_drive + age:doors + age:fuel + age:car_type +
##     age:gearbox + car_mileage:post_info + car_mileage:A_C + car_mileage:seats_amount +
##     car_mileage:color + car_mileage:type_of_drive + car_mileage:doors +
##     car_mileage:fuel + car_mileage:car_type + car_mileage:gearbox +
##     engine_capacity:post_info + engine_capacity:A_C + engine_capacity:seats_amount +
##     engine_capacity:color + engine_capacity:type_of_drive + engine_capacity:doors +
##     engine_capacity:fuel + engine_capacity:car_type + engine_capacity:gearbox +
##     horsepower:post_info + horsepower:A_C + horsepower:seats_amount +
##     horsepower:color + horsepower:type_of_drive + horsepower:doors +
##     horsepower:fuel + horsepower:car_type + horsepower:gearbox
##
## Final Model:
## price ~ age + car_mileage + engine_capacity + horsepower + post_info +
##     A_C + seats_amount + color + type_of_drive + doors + fuel +
```

```
##     car_type + gearbox + age:A_C + age:seats_amount + age:color +
##     age:type_of_drive + age:doors + age:fuel + age:car_type +
##     age:gearbox + car_mileage:post_info + car_mileage:A_C + car_mileage:seats_amount +
##     car_mileage:type_of_drive + car_mileage:doors + car_mileage:fuel +
##     car_mileage:car_type + car_mileage:gearbox + engine_capacity:post_info +
##     engine_capacity:A_C + engine_capacity:seats_amount + engine_capacity:type_of_drive +
##     engine_capacity:fuel + engine_capacity:car_type + engine_capacity:gearbox +
##     horsepower:post_info + horsepower:seats_amount + horsepower:color +
##     horsepower:fuel + horsepower:car_type + horsepower:gearbox
##
##
##                         Step Df    Deviance Resid. Df  Resid. Dev      AIC
## 1                                                6727 50752178061 112398.0
## 2 - horsepower:type_of_drive  2    489335.8      6729 50752667397 112394.1
## 3           - horsepower:A_C  2  16033793.4      6731 50768701190 112392.3
## 4    - engine_capacity:color 17 230316433.5      6748 50999017624 112390.3
## 5        - car_mileage:color 17 210007501.8      6765 51209025126 112385.4
## 6    - engine_capacity:doors  1   3790831.3      6766 51212815957 112383.9
```

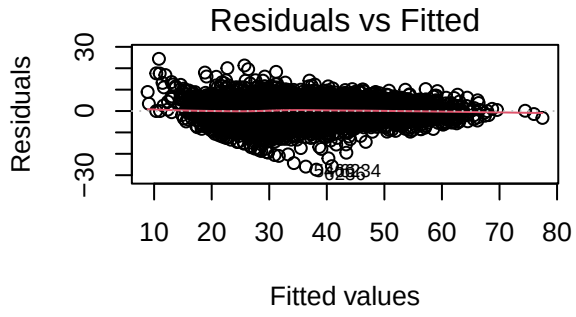
```
## lm(formula = price ~ age + car_mileage + engine_capacity + horsepower +
##     post_info + A_C + seats_amount + color + type_of_drive +
##     doors + fuel + car_type + gearbox + age:A_C + age:seats_amount +
##     age:color + age:type_of_drive + age:doors + age:fuel + age:car_type +
##     age:gearbox + car_mileage:post_info + car_mileage:A_C + car_mileage:seats_amount +
##     car_mileage:type_of_drive + car_mileage:doors + car_mileage:fuel +
##     car_mileage:car_type + car_mileage:gearbox + engine_capacity:post_info +
##     engine_capacity:A_C + engine_capacity:seats_amount + engine_capacity:type_of_drive +
##     engine_capacity:fuel + engine_capacity:car_type + engine_capacity:gearbox +
##     horsepower:post_info + horsepower:seats_amount + horsepower:color +
##     horsepower:fuel + horsepower:car_type + horsepower:gearbox,
##     data = clean_data)
```

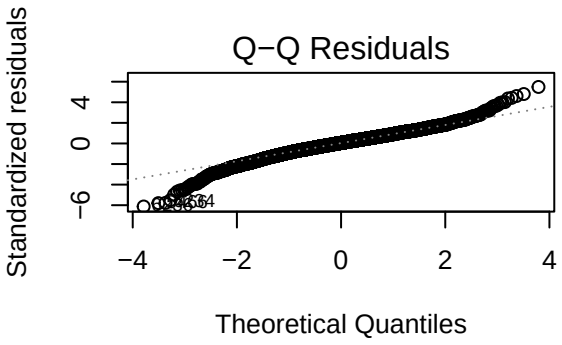
It seems that some line assumptions maybe violated, so let's perform a Box-Cox transformation.

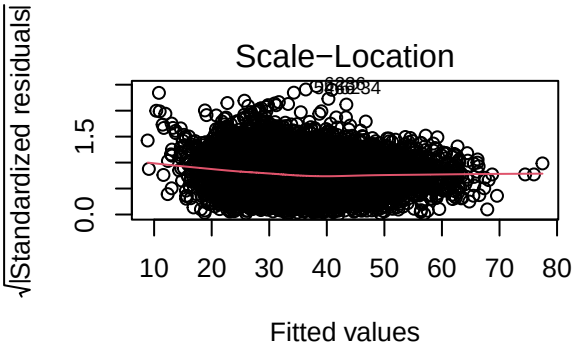
Model After Box-Cox Transformation

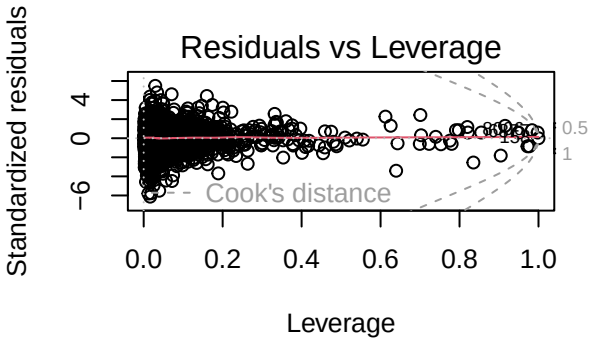
```
## Best lambda: 0.3
```

```
## Applied Box-Cox transformation with lambda = 0.3 to `price`.
```



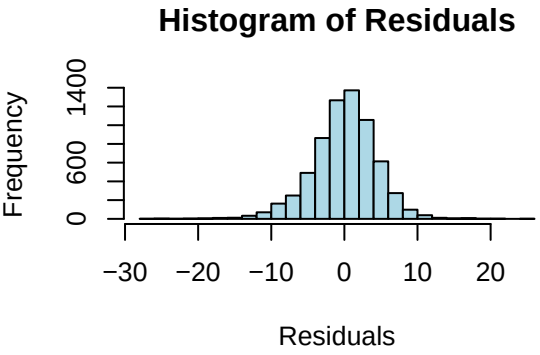


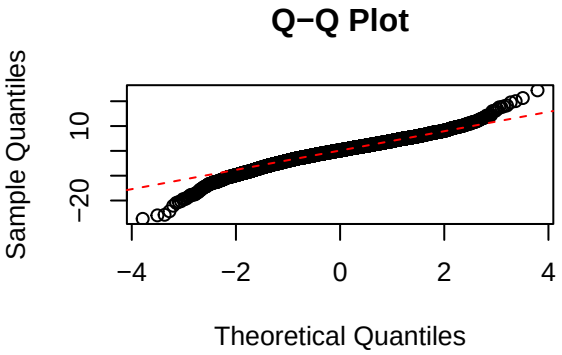





```
##
##  studentized Breusch-Pagan test
##
## data:  final_model
## BP = 556.24, df = 246, p-value < 2.2e-16

## summary of model (R^2, adj-R^2): 0.819821 0.8129008
```






```
Residuals:
    Min      1Q  Median      3Q     Max
-51.171  -6.255   0.217   6.665  40.874

Residual standard error: 10.53 on 5799 degrees of freedom
Multiple R-squared:  0.8711,    Adjusted R-squared:  0.866
F-statistic: 169.6 on 231 and 5799 DF,  p-value: < 2.2e-16
```

We ran VIF to check for multicolinearity (high VIF could imply unreliable estimates), AIC to remove unnecessary parameters, and we made three models: main effect, model with interactions, and transformed interaction models. the final model had the highest coefficient of determination. we did a transformation from interaction model as we realized that the normality assumption may be violated based off the second model's qq-plot. The second model had

a higher R-squared value, which is a good indicator that the interaction terms improved model, it explained ~74% of variability in Price. These R^2 values indicate that our model captures about 81% of variability in Price after Box-Cox tansformation, which is better.

Using our hold-out validation set, we can perform a little bit more verification:

```
## [1] 1337
```

```
## [1] 44756721
```

```
## [1] 19.57921
```

We used hold-out validation on our data: Trained 6652/7988 (83.3%) of the data Tested on 1336/7988 (16.7%) of the data The standard error of the data: Standard Error of Training Data: ~3 This is expected, since it should almost perfectly align with what we have in training data (except for errors caused by lack of emission_class) Standard Error of Testing Data: ~6,655 Means that our prediction is only about $6655 off of the actual price on average (not accounting for errors caused by lack of emission_class), meaning that it's fairly accurate

WLS is a next step since we failed homoscedasticity (equal variance based off bp test) assumption. This means that we'd include a weight vector when fitting the model, and hopefully this would fix the equal variance problem.

Therefore, the most influential factors are: Age, Horsepower, Car Mileage, Engine Capacity. More factors in the data collection stage could help us improve our model. Various visual or mechanical problems of the car might have a big influence and is missing from our data.

References used:

Mmakarovlab. (2024). Car price prediction 2024 (fresh market posts). Retrieved from https://www.kaggle.com/datasets/mmakarovlab/serbia-car-sales-prices?resource=download